

Bayesian Protein Sequence and Structure Alignment

Christopher J. Fallaize^{1*}, Peter J. Green^{2,3}, Kanti V. Mardia^{4,5} and Stuart Barber⁴

¹ School of Mathematical Sciences, University of Nottingham

² School of Mathematics, University of Bristol

³ School of Mathematical Sciences, University of Technology, Sydney

⁴ Department of Statistics, School of Mathematics, University of Leeds

⁵ Department of Statistics, University of Oxford

Abstract

One of the major problems in biology is related to protein folding. The folding process is known to depend on both the protein's sequence (1-D) and structure (3-D). Both these properties need to be considered when aligning two proteins; they are also influenced by the evolutionary distance between the proteins to be aligned. We propose a Bayesian method to align proteins using both the sequence and 3-D structure of the proteins. The problem involves what are known as “gaps” in the sequence, which we incorporate in our model, and an MCMC implementation is provided. Also, we show that the procedure can be used to give insight into evolutionary distances between proteins. Various illustrative examples from protein bioinformatics are studied.

Keywords: Evolutionary distance, gap penalty prior, Markov chain Monte Carlo, structural bioinformatics, unlabelled shape analysis.

1 Introduction

In this paper we consider the alignment of a pair of proteins using their 3-dimensional structures. When aligning proteins, one aim is to determine whether they are related, in the sense that they have evolved from a common ancestor. At the primary level, a protein is a sequence of letters, with each letter representing the amino acid residue at the corresponding position in the sequence. Therefore, a measure of how closely a pair of proteins are related may be obtained by aligning their sequences as closely as possible and assessing how well the two sequences complement each other after alignment. However, the structure of a protein is more conserved than its sequence. Over a period of evolution, the sequence of a protein may change through substitutions of amino acid residues from one type into another at a particular position, from the insertion of new amino acid residues, or from deletion of existing residues. However, the overall physical structure may remain essentially unchanged, at least in regions of the protein which are functionally important. Therefore, a better measure of how closely two proteins are related can be obtained from aligning their structures, and

*Author for correspondence: Chris.Fallaize@nottingham.ac.uk

with the increasing number of protein structures becoming available and deposited in databases such as the Protein Data Bank (Berman et al., 2000), reliable methods for protein structure alignment are becoming more important. Many methods for doing so have been developed, such as DALI (Holm and Sander, 1993), CE (Shindyalov and Bourne, 1998), LGA (Zemla, 2003), SSAP (Orengo and Taylor, 1996), MAMMOTH (Ortiz et al., 2002) and others. These methods are based on computational algorithms designed to find an optimal alignment in some sense, and do not give any indication of uncertainty in this optimum; for instance there may be high uncertainty in some areas of the alignment, and other areas where the alignment between the two structures is very good. Therefore, there is a need for probabilistic methods which allow uncertainty in the alignment to be quantified.

Mathematically, a protein can be represented as a configuration of m points, $\{\mathbf{x}_j\}_{j=1}^m$, $\mathbf{x}_j \in \mathbb{R}^3$. For example, the points often represent the locations of the C_α (alpha-carbon) atoms. The problem is then to align this configuration with that of another protein $\{\mathbf{y}_k\}_{k=1}^n$. That is, we seek a rigid body transformation of the y points such that

$$\mathbf{A}\mathbf{y} + \boldsymbol{\tau} = \mathbf{x},$$

for any pair of matching points \mathbf{x} and \mathbf{y} . Here, \mathbf{A} is a 3×3 rotation matrix and $\boldsymbol{\tau} \in \mathbb{R}^3$ is a translation vector. The correspondence between points on the two configurations is encoded in an $m \times n$ matrix \mathbf{M} , with elements M_{jk} , where

$$M_{jk} = \begin{cases} 1, & \text{if } \mathbf{x}_j \text{ and } \mathbf{y}_k \text{ are matched,} \\ 0, & \text{otherwise.} \end{cases}$$

Usually, \mathbf{M} is not known and it is the main object of interest about which to draw inference; this is known as unlabelled shape analysis, and the problem of protein structure alignment is an important example of this.

Unlabelled shape analysis has been the focus of much recent research interest in statistical shape analysis, motivated by important applications such as that of protein structure alignment. Here, we consider a Bayesian solution to the problem. From the Bayesian viewpoint, there are essentially two approaches that have been developed for unlabelled shape analysis. One approach is to maximize over the transformation parameters \mathbf{A} and $\boldsymbol{\tau}$ (Dryden et al., 2007; Rodriguez and Schmidler, 2010; Schmidler, 2007) which can be viewed as using a Laplace approximation to integrate out \mathbf{A} and $\boldsymbol{\tau}$ and using the marginal posterior distribution for inference about M (Kenobi and Dryden, 2012). An alternative approach is to consider a fully Bayesian model, where the transformation parameters are included as unknown parameters in the model about which to draw inference (Green and Mardia, 2006). In this manner, uncertainty in these parameters is accounted for and correctly propagated throughout the analysis (Wilkinson, 2007). The underlying formulation is very flexible. For instance, Green and Mardia (2006) considered rigid body transformations in their applications, but Mardia et al. (2013) demonstrated applications using full similarity transformations. Forbes and Lauritzen (2013) also use this approach with similarity transformations in the context of finger-

print matching, an important application in forensic science, which demonstrates the wide-ranging applicability of the underlying methodology.

In this paper, we consider the alignment of protein structures within the fully Bayesian framework of Green and Mardia (2006), but with an important change to the prior model for the matching matrix \mathbf{M} . In the original setting of Green and Mardia (2006), conditional on the total number of matched points L , then every possible \mathbf{M} consistent with L matched points was considered equally likely. When aligning homologous proteins which are thought to have evolved from a common ancestor, it is important to preserve the sequence order of the points in the matching given by \mathbf{M} . Therefore, we require a prior for \mathbf{M} which imposes this constraint. Here, we use the prior suggested by Rodriguez and Schmidler (2010), and show how this can be incorporated into the fully Bayesian model.

The paper is structured as follows. In Section 2, we describe the alignment of protein sequences, which allows us to illustrate the concept of an alignment and introduce some important concepts which are used frequently in the subsequent discussion of structural alignment, in particular the concept of a gap in an alignment. In Section 3, we describe the Bayesian model for structural alignment, and give details of the prior for \mathbf{M} which satisfies the sequence order constraint. In Section 4 we apply our method to two challenging examples considered previously in the literature. We illustrate how amino acid sequence information can be incorporated into the model in Section 5, and show how this allows a measure of evolutionary distance between the proteins to be obtained. We conclude the paper with a discussion.

2 Sequence alignment

Consider a pair of sequences $S^x = \{s_j^x\}_{j=1}^m$ and $S^y = \{s_k^y\}_{k=1}^n$, with elements $s_j^x, s_k^y \in \mathcal{S}$, where \mathcal{S} is the set of 20 letters representing the 20 amino acids. Therefore, each sequence is a string of letters, with each letter representing the amino acid type at the corresponding position on the protein. Pairwise sequence alignment algorithms seek to align the two sequences as closely as possible according to some scoring mechanism. Each pair of aligned residues is given a score based on how well the two corresponding amino acid types complement each other; informally, alignments between the same amino acid, or amino acids with similar properties, achieve favourable scores, and those between amino acids with different properties achieve less favourable scores. Figure 1 (a) shows an example of an alignment between two sequences. In some positions, there is an alignment between identical amino acid types, and in other positions different amino acid types are aligned. Methods for scoring matches between different amino acid types have been developed, such as the PAM (Dayhoff et al., 1978) and BLOSUM (Henikoff and Henikoff, 1992) matrices. These are 20×20 symmetrical matrices, with entries giving scores for matches between any pair of amino acid types. The entries are usually expressed as log odds scores, so that the total score of an alignment is a sum of the scores of each aligned pair. Pairs of amino acid types which are most compatible

biologically have high positive scores, and those with very different properties have negative scores; a good overall alignment will therefore have a high score. The scores can then be tested for statistical significance, with a statistically significant score providing evidence against the null hypothesis that the sequences could have been observed by chance (see Durbin et al. (1998) for more details).

S^x		G	K	S	T	L	L	K	K	L		
S^y		G	K	G	T	I	C	K	A	L		
(a)												
S^x		H	E	A	G	A	W	G	H	E	E	
S^y		P	-	-	-	A	W	H	E	A	E	
(b)												
S^x		H	E	A	G	A	W	G	H	E	-	E
S^y		-	P	-	-	A	W	-	H	E	A	E
(c)												

Figure 1: Three examples of a sequence alignment. In (a), there are no gaps. In (b) and (c) there are gaps in one or both sequences, and of different lengths. Alignments (b) and (c) are from Durbin et al. (1998, p. 91).

Usually, in order to achieve the best possible scoring alignment, it is necessary to insert gaps in one or both of the sequences. Figure 1 (b) shows an alignment with gaps in one of the sequences. The gaps allow alignments between high scoring pairs of residues to be made. Over the course of evolution, extra residues may be inserted in one sequence or deleted from the other sequence; such instances are referred to as indels. Additionally, a mutation could occur in one or both of the sequences at a certain position, such that the amino acid at that position is substituted for one of a different type. As such, a pair of sequences which have evolved from a common ancestor may have been subject to many insertions, deletions and substitutions; they may contain regions which have remained largely conserved, and other regions which have diverged quite substantially. It is the goal of sequence alignment to ensure that the highly conserved regions are aligned, and that regions or pairs of residues which are no longer related are not. Gaps enable this to be achieved, and can be interpreted as an explanation for indels which result in the observed differences between the sequences. Figure 1 (c) shows an alignment with gaps in both sequences, allowing alignments between a larger number of identical amino acid types in the second part of the alignment.

3 Bayesian structure alignment

We now describe the Bayesian model for protein structure alignment. A new prior for the matching matrix \mathbf{M} is proposed, based on a penalty function for the number

and length of gaps in the alignment implied by \mathbf{M} ; this prior is referred to as a gap prior, and it imposes the new constraint that sequence order must be preserved in any alignment.

3.1 Likelihood

We have two point configurations, $\{\mathbf{x}\}$ and $\{\mathbf{y}\}$, consisting of m and n points respectively. The points are labelled \mathbf{x}_j , $j = 1, \dots, m$ and \mathbf{y}_k , $k = 1, \dots, n$, where $\mathbf{x}_j, \mathbf{y}_k \in \mathbb{R}^d$; in our case, protein structures are 3-dimensional configurations and $d = 3$. A rigid body transformation which transforms points on $\{\mathbf{y}\}$ into x -space is of the form $\mathbf{A}\mathbf{y} + \boldsymbol{\tau}$, where \mathbf{A} is a $d \times d$ rotation matrix and $\boldsymbol{\tau} \in \mathbb{R}^d$ is a translation vector. As in Green and Mardia (2006), we have

$$\begin{aligned} \mathbf{x}_j &= \boldsymbol{\mu}_{\xi_j} + \boldsymbol{\epsilon}_j & j = 1, \dots, m, \\ \mathbf{A}\mathbf{y}_k + \boldsymbol{\tau} &= \boldsymbol{\mu}_{\eta_k} + \boldsymbol{\epsilon}_k & k = 1, \dots, n, \end{aligned}$$

where $\{\boldsymbol{\mu}\}$ is an unobserved hidden configuration, from which the observed points are derived. The $\boldsymbol{\epsilon}$ terms represent error in the observed points, which are regarded as noisy observations of the true locations on $\{\boldsymbol{\mu}\}$. Here, we use a spherical Gaussian model for the errors, so that $\boldsymbol{\epsilon} \sim N_d(0, \sigma^2 \mathbf{I})$, where \mathbf{I} is the $d \times d$ identity matrix; the parameter σ^2 therefore represents the error variance. The ξ and η terms give the mapping between points on $\{\boldsymbol{\mu}\}$ and points on $\{\mathbf{x}\}$ and $\{\mathbf{y}\}$ respectively. In particular, when $\xi_j = \eta_k$ then the corresponding x and y points are both realisations of the same hidden location, and are regarded as matched points. The matching between the configurations is captured by the matching matrix \mathbf{M} . We impose the constraint that a given point on one configuration can match at most one point on the other configuration, so that each row or column of \mathbf{M} has at most one non-zero entry. Then, $\sum_{j,k} M_{jk} = L$, where L is the

total number of matched points.

The points on $\{\boldsymbol{\mu}\}$ are assumed to form a homogeneous Poisson process over a region of volume v , and these hidden points can be integrated out. Then, assuming v is large relative to the support of the density of the error terms, the (approximate) respective likelihood contributions of the unmatched x , unmatched y and matched points are

$$v^{-(m-L)}, (|\mathbf{A}|/v)^{n-L}, (|\mathbf{A}|/v)^L \prod_{j,k:M_{jk}=1} \frac{\phi\{(\mathbf{x}_j - \mathbf{A}\mathbf{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d},$$

where $\phi(\cdot)$ is the d -dimensional standard normal density. Hence the likelihood of the observed data given \mathbf{M} (and the other parameters) is

$$p(\mathbf{x}, \mathbf{y} | \mathbf{M}, \mathbf{A}, \boldsymbol{\tau}, \sigma) = v^{-(m+n-L)} |\mathbf{A}|^n \prod_{j,k:M_{jk}=1} \frac{\phi\{(\mathbf{x}_j - \mathbf{A}\mathbf{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}. \quad (1)$$

3.2 Gap prior

Recall that our main objective is to align two configurations when the points on each configuration have a meaningful ordering which must be preserved in any resulting alignment. To obtain the best possible structural alignment, it may be necessary to insert gaps in the corresponding sequence alignment in one or both of the sequences. We summarise an alignment with the matching matrix \mathbf{M} , for which we use a prior which imposes the sequence order constraint. Specifically, we use the prior

$$p(\mathbf{M}; g, h) = Z(g, h) \exp\{-u(\mathbf{M}; g, h)\}, \quad (2)$$

as in Rodriguez and Schmidler (2010), where $u(\mathbf{M}; g, h)$ is a penalty function which penalises gaps in the alignment, and $Z(g, h)$ is a normalising constant. The parameters g and h are gap opening and extension penalties respectively. The penalty function is

$$u(\mathbf{M}; g, h) = gS(\mathbf{M}) + h \sum_{i=1}^{S(\mathbf{M})} (l_i - 1),$$

where $S(\mathbf{M})$ is the number of instances where a new gap in the alignment is opened, and l_i is the length of the i th gap. To illustrate what is meant by a new gap and length of a gap, consider again the sequence alignment in Figure 1 (b). In the first sequence, the second residue is not matched to a residue on the second sequence; instead it is aligned to a “-”, indicating that a gap has been opened. That is, a gap opening is said to have been created where a residue in one sequence is unmatched, but the previous residue in the same sequence was aligned to a residue in the other sequence. The length of the gap is then the number of unmatched residues (in the same sequence) until another matched pair; therefore, the gap in Figure 1 (b) is of length 3.

In Figure 1 (c), the first sequence has one gap, of length 1, and the second sequence has three gaps, of lengths 1, 2 and 1. Note that the two sequences are considered independently when counting the number and length of the gaps, so that a gap in one sequence followed immediately by a gap in the other sequence would be counted as two different gaps.

We now formalise the notation in our structural alignment setting. Suppose we have configurations \mathbf{X} and \mathbf{Y} , consisting of m and n points respectively, with L matched points between the two. Further, suppose the indices of the matched points on \mathbf{X} are $j_1 < j_2 < \dots < j_L$ and the indices of the matched points on \mathbf{Y} are $k_1 < k_2 < \dots < k_L$. Then the total penalty is

$$\sum_{i=0}^L f(j_{i+1} - j_i) + \sum_{i=0}^L f(k_{i+1} - k_i),$$

where

$$f(r) = \begin{cases} 0 & r = 1 \\ g & r = 2 \\ g + (r - 2)h & r > 2. \end{cases}$$

We set $j_0 = k_0 = 0$ and $j_{L+1} = m$, $k_{L+1} = n$ to account for the start and end points of the sequences. Note that using this formulation, other penalty functions could be incorporated in the same way. In particular, any function which can be decomposed into a sum of penalty contributions from each pair of matching indices could be used. The gap penalty function we have used here is of a form widely used in sequence alignment (Durbin et al., 1998). It is common to penalise gap openings more than extensions, i.e. set $g > h$.

The new prior on \mathbf{M} results in a change in the joint model (Equation (6) of Green and Mardia (2006)). Multiplying (1) and (2) we obtain

$$p(\mathbf{M}, \mathbf{x}, \mathbf{y} | \mathbf{A}, \boldsymbol{\tau}, \sigma) \propto |\mathbf{A}|^n v^L \exp\{-U(\mathbf{M}; g, h)\} \prod_{j,k: M_{jk}=1} \frac{\phi\{(\mathbf{x}_j - \mathbf{A}\mathbf{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}$$

and the joint model is

$$\begin{aligned} p(\mathbf{M}, \mathbf{A}, \boldsymbol{\tau}, \sigma, \mathbf{x}, \mathbf{y}) &\propto p(\mathbf{A})p(\boldsymbol{\tau})p(\sigma)|\mathbf{A}|^n v^L \exp\{-U(\mathbf{M}; g, h)\} \\ &\times \prod_{j,k: M_{jk}=1} \frac{\phi\{(\mathbf{x}_j - \mathbf{A}\mathbf{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}, \end{aligned} \quad (3)$$

where $p(\mathbf{A})$, $p(\boldsymbol{\tau})$ and $p(\sigma)$ are the prior distributions of \mathbf{A} , $\boldsymbol{\tau}$ and σ respectively. The rotation matrix \mathbf{A} has a matrix-Fisher prior distribution, where $p(\mathbf{A}) \propto \exp\{\text{tr}(\mathbf{F}_0^T \mathbf{A})\}$ and the parameter \mathbf{F}_0 is a $d \times d$ matrix. \mathbf{A} is parameterized by Eulerian angles, $\theta_{12}, \theta_{13}, \theta_{23}$, say, in the case $d = 3$. In our examples we use a uniform prior on \mathbf{A} , which is the special case where \mathbf{F}_0 is the $d \times d$ matrix of zeroes. \mathbf{A} then has a uniform prior with respect to the invariant measure on $SO(3)$, the Haar measure, where $SO(3)$ is the special orthogonal group of all $d \times d$ rotation matrices. For the translation vector $\boldsymbol{\tau}$, we have $\boldsymbol{\tau} \sim N_d(\boldsymbol{\mu}_\tau, \sigma_\tau^2 \mathbf{I}_d)$, where $\boldsymbol{\mu}_\tau$ is a mean vector and $\sigma_\tau^2 \mathbf{I}_d$ a covariance matrix, with \mathbf{I}_d the $d \times d$ identity matrix. For the noise parameter σ , we have $\sigma^{-2} \sim \Gamma(\alpha, \beta)$, so $p(\sigma^{-2}) \propto \sigma^{-2(\alpha+1)} \exp(-\frac{\beta}{\sigma^2})$.

Note that the volume term v is now no longer absorbed into the normalising constant, unlike in the model of Green and Mardia (2006), where this term cancelled with a corresponding term from the prior for \mathbf{M} . We discuss sensitivity to user-specified values of v in Section 4.3.

3.3 Sampling \mathbf{M}

Updates for the parameters \mathbf{A} , $\boldsymbol{\tau}$ and σ are as in Green and Mardia (2006). We now describe the mechanism for generating posterior samples of \mathbf{M} , using Metropolis-

Hastings updates. Suppose our current alignment is \mathbf{M} , and we have a proposal value \mathbf{M}' drawn from a proposal density $q(\mathbf{M}'; \mathbf{M})$. Then the acceptance probability is

$$\alpha = \min \left\{ 1, \frac{p(\mathbf{M}', \mathbf{A}, \boldsymbol{\tau}, \sigma, \mathbf{x}, \mathbf{y})q(\mathbf{M}; \mathbf{M}')}{p(\mathbf{M}, \mathbf{A}, \boldsymbol{\tau}, \sigma, \mathbf{x}, \mathbf{y})q(\mathbf{M}'; \mathbf{M})} \right\},$$

where $p(\cdot)$ is the joint model (3).

Suppose there are currently L matches with indices $j_1 < j_2 < \dots < j_L$ and $k_1 < k_2 < \dots < k_L$. Suppose further that we propose to add a match (j^*, k^*) , where $j_i < j^* < j_{i+1}$ and $k_i < k^* < k_{i+1}$, $i = 0, \dots, L$, and we have $j_0 = k_0 = 0$ and $j_{L+1} = m + 1, k_{L+1} = n + 1$. We have

$$\frac{p(\mathbf{M}', \mathbf{A}, \boldsymbol{\tau}, \sigma, \mathbf{x}, \mathbf{y})}{p(\mathbf{M}, \mathbf{A}, \boldsymbol{\tau}, \sigma, \mathbf{x}, \mathbf{y})} = \exp\{U(\mathbf{M}) - U(\mathbf{M}')\} \times \frac{v\phi\{(\mathbf{x}_{j^*} - \mathbf{A}\mathbf{y}_{k^*} - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d},$$

where $U(\mathbf{M}) - U(\mathbf{M}')$ is the reduction in the gap penalty achieved by adding the match (j^*, k^*) ; this comprises two terms, one from the j indices and one from the k indices. Specifically, the reduction in penalty arising from the j terms (assuming $j_{i+1} - j_i \geq 2$) has the following form:

$$r(j^*, j_i, j_{i+1}, g, h) = \begin{cases} g & \text{if } j_{i+1} - j_i = 2 \\ h & \text{if } j_{i+1} - j_i > 2 \text{ and } \{j^* = (j_i + 1) \text{ or } (j_{i+1} - 1)\} \\ 2h - g & \text{if } j_{i+1} - j_i > 2 \text{ and } \{j^* \neq (j_i + 1) \text{ or } (j_{i+1} - 1)\}. \end{cases}$$

The reduction from the k indices has the same form. So we have a total reduction of

$$U(\mathbf{M}) - U(\mathbf{M}') = r(j^*, j_i, j_{i+1}, g, h) + r(k^*, k_i, k_{i+1}, g, h).$$

We now consider how to obtain a proposal \mathbf{M}' given the current alignment \mathbf{M} . We select a point uniformly at random from the $m+n$ points $\mathbf{x}_1, \dots, \mathbf{x}_m, \mathbf{y}_1, \dots, \mathbf{y}_n$. Suppose we select a point on configuration $\{\mathbf{x}\}$, \mathbf{x}_j say. (By symmetry, the process is similar if a point on $\{\mathbf{y}\}$ is selected.) If \mathbf{x}_j is currently unmatched, then we propose adding a match between \mathbf{x}_j and another currently-unmatched point, \mathbf{y}_k say. If, however, \mathbf{x}_j is currently matched, to a point \mathbf{y}_k say, then with probability p^* we propose deleting this match, and with probability $(1 - p^*)$ we propose switching this to a match between \mathbf{x}_j and another unmatched point \mathbf{y}_{k_2} . We now describe each of these three types of proposal in more detail.

Case 1: \mathbf{x}_j is currently unmatched, with $j_i < j < j_{i+1}$ (where j_i and j_{i+1} are the nearest matched points either side of \mathbf{x}_j , matching points on configuration $\{\mathbf{y}\}$ with indices k_i and k_{i+1} respectively, as above.) Note that if $k_{i+1} - k_i = 1$, there is no space for a new match. If however $k_{i+1} - k_i \geq 2$ then propose a match between \mathbf{x}_j and \mathbf{y}_k , with the index k drawn uniformly from the interval (k_i, k_{i+1}) , so each possible index k has a probability $1/(k_{i+1} - k_i - 1)$ of being selected. Then $q(\mathbf{M}'; \mathbf{M}) = 1/(k_{i+1} - k_i - 1)$. Also, $q(\mathbf{M}; \mathbf{M}') = p^*$, where p^* is the probability of proposing a deletion of a matched

point. So we have acceptance probability

$$\alpha = \min \left[1, \exp\{r(j, j_i, j_{i+1}, g, h) + r(k, k_i, k_{i+1}, g, h)\} \times p^* \right. \\ \left. \times (k_{i+1} - k_i - 1) \times \frac{v\phi\{(\mathbf{x}_j - \mathbf{A}\mathbf{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d} \right].$$

Case 2: \mathbf{x}_j is currently matched (to \mathbf{y}_k say) and we propose deletion with probability p^* . Therefore, $q(\mathbf{M}'; \mathbf{M}) = p^*$. Also, $q(\mathbf{M}; \mathbf{M}') = 1/(k_{i+1} - k_i - 1)$, arising from adding the match (j, k) and using the same argument as in case 1. The reduction in penalty from deleting the match is equal to the negative of the reduction that would be obtained from adding the match, i.e the deletion is accepted with probability

$$\alpha = \min \left[1, \exp\{-r(j, j_i, j_{i+1}, g, h) - r(k, k_i, k_{i+1}, g, h)\} / \{p^* \times (k_{i+1} - k_i - 1)\} \right. \\ \left. \times \frac{(\sigma\sqrt{2})^d}{v\phi\{(\mathbf{x}_j - \mathbf{A}\mathbf{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}} \right].$$

Case 3: As in case 2, \mathbf{x}_j is currently matched to \mathbf{y}_k and we propose switching to match \mathbf{y}_{k_2} , with k_2 drawn uniformly from the interval (k_i, k_{i+1}) , where k_i and k_{i+1} are the nearest matched points either side of \mathbf{y}_k . Again note that if $k_{i+1} - k_i = 1$ there is no possible match to switch to and the current match is retained. Otherwise, the reduction in the penalty term from switching \mathbf{x}_j to match \mathbf{y}_k is $r(k_2, k_i, k_{i+1}, g, h) - r(k, k_i, k_{i+1}, g, h)$, where the first term is the reduction arising from adding the match with \mathbf{y}_{k_2} and the second term is the reduction due to deleting a match with \mathbf{y}_k (i.e the negative of the reduction due to adding this match). In this case there is no term from the j indices since \mathbf{x}_j remains matched before and after. Also, we have $q(\mathbf{M}; \mathbf{M}') = q(\mathbf{M}'; \mathbf{M}) = (1 - p^*)(k_{i+1} - k_i - 1)$. Hence, the proposed switch is accepted with probability

$$\alpha = \min \left[1, \exp\{r(k_2, k_i, k_{i+1}, g, h) - r(k, k_i, k_{i+1}, g, h)\} \times \frac{\phi\{(\mathbf{x}_j - \mathbf{A}\mathbf{y}_{k_2} - \boldsymbol{\tau})\}}{\phi\{(\mathbf{x}_j - \mathbf{A}\mathbf{y}_k - \boldsymbol{\tau})\}} \right].$$

Note therefore that under this sampling method, we make only small perturbations to the alignment at each iteration, by either removing a match, adding a match, or switching a match, so that the total number of matches can change by at most 1. As the changes are small, we may typically want to propose a number of changes to \mathbf{M} per iteration of the MCMC sampler.

This method of sampling \mathbf{M} from the posterior distribution is quite different to the method used by Rodriguez and Schmidler (2010). These authors sample a whole new matching matrix \mathbf{M} at each iteration of the MCMC sampler, using dynamic programming recursions analogous to those used in sequence alignment algorithms. Therefore, generating proposal moves for \mathbf{M} is more computationally intensive than in our case, but the sampler may converge to equilibrium faster than our method which uses local moves, so that fewer iterations of the sampler are needed overall.

3.4 Gap penalty parameters as unknowns

Previously, we have assumed the parameters g and h are fixed by the user. We now consider including them as unknowns, and detail two approaches to dealing with this situation. The first is to sample them from the joint posterior distribution, and the second is to integrate them out and work with the marginal prior distribution of \mathbf{M} .

3.4.1 Inference for the gap penalty parameters

Rather than treating the gap penalty parameters as fixed, we can treat them as unknowns and sample them from the posterior distribution in addition to the other parameters we are already sampling. Note that the penalty parameters g and h are non-negative, and for convenience we place gamma prior distributions on them, with $g \sim \Gamma(a_g, b_g)$, $h \sim \Gamma(a_h, b_h)$, so $p(g) \propto g^{a_g-1} \exp(-b_g g)$, and similarly for h . The joint posterior is now given by

$$p(\mathbf{M}, \mathbf{A}, \boldsymbol{\tau}, \sigma, g, h, \mathbf{x}, \mathbf{y}) = p(\mathbf{A}, \boldsymbol{\tau}, \sigma, \mathbf{x}, \mathbf{y} | \mathbf{M}) p(\mathbf{M} | g, h) p(g) p(h),$$

leading to

$$\begin{aligned} p(\mathbf{M}, \mathbf{A}, \boldsymbol{\tau}, \sigma, g, h, \mathbf{x}, \mathbf{y}) &\propto p(\mathbf{A}) p(\boldsymbol{\tau}) p(\sigma) |\mathbf{A}|^n v^L g^{a_g-1} \exp(-b_g g) h^{a_h-1} \exp(-b_h h) Z(g, h) \\ &\times \exp\{-U(\mathbf{M}; g, h)\} \prod_{j,k:M_{jk}=1} \frac{\phi\{(\mathbf{x}_j - \mathbf{A}\mathbf{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}. \end{aligned} \quad (4)$$

The parameters g and h are sampled using Metropolis-Hastings updates, using a geometric random walk with uniform perturbations to generate proposals. For example, given the current value g for the gap opening penalty parameter, a value g' is proposed, where $g' = g \exp(u)$ and u is a uniform random variable on the interval $[-a, a]$, $a > 0$. The proposal density is $q(g'; g) = \frac{1}{2ag'}$, leading to the acceptance probability

$$\alpha_g = \min \left\{ 1, \frac{Z(g', h) \exp\{-U(\mathbf{M}; g', h)\} g'^{a_g} \exp(-a_g g')}{Z(g, h) \exp\{-U(\mathbf{M}; g, h)\} g^{a_g} \exp(-a_g g)} \right\}.$$

The normalising constant $Z(g, h)$ is found by summing the prior distribution (2) over all possible matching matrices; that is, $Z(g, h)^{-1} = \sum_{\mathbf{M}'} \exp\{-U(\mathbf{M}'; g, h)\}$. This summation can be computed using dynamic programming recursions, as given in Appendix A of Rodriguez and Schmidler (2010). The gap extension parameter h is updated in a similar manner.

3.4.2 Integrating out the gap penalty parameters

An alternative approach is to integrate out the parameters g and h , and work directly with the marginal prior $p(\mathbf{M})$, an approach adopted by Liu and Lawrence (1999) in the sequence alignment setting. The joint prior distribution is

$$p(\mathbf{M}, g, h) = Z(g, h) \exp\{-gS(\mathbf{M}) - hL(\mathbf{M})\} g^{a_g-1} \exp(-b_g g) h^{a_h-1} \exp(-b_h h) \frac{b_g^{a_g} b_h^{a_h}}{\Gamma(a_g)\Gamma(a_h)}.$$

Recall that $S(\mathbf{M})$ is the total number of gap openings, and we now write $L(\mathbf{M}) = \sum_{i=1}^{S(\mathbf{M})} (l_i - 1)$, so that $L(\mathbf{M})$ is the total number of gap extensions. Then the marginal prior distribution of \mathbf{M} is

$$\begin{aligned} p(\mathbf{M}) &= \int \int Z(g, h) \exp\{-gS(\mathbf{M}) - hL(\mathbf{M})\} g^{a_g-1} \exp(-b_g g) h^{a_h-1} \exp(-b_h h) \frac{b_g^{a_g} b_h^{a_h}}{\Gamma(a_g)\Gamma(a_h)} dg dh, \\ &= \frac{b_g^{a_g} b_h^{a_h}}{\Gamma(a_g)\Gamma(a_h)} \int \int \exp\{\psi(g, h)\} dg dh, \end{aligned}$$

where

$$\psi(g, h) = \log(Z(g, h)) - g(S(\mathbf{M}) + b_g) - h(L(\mathbf{M}) + b_h) + (a_g - 1) \log(g) + (a_h - 1) \log(h).$$

We perform this integral using a mid-point quadrature rule based on N^2 grid points:

$$P(\mathbf{M}) \approx \Delta_g \Delta_h \sum_{i=1}^N \sum_{j=1}^N \exp\{\psi(g_i, h_j)\},$$

where $g_i = g_0 + (i - 0.5)\Delta_g$ and $h_j = h_0 + (j - 0.5)\Delta_h$, $i, j = 1, \dots, N$, where Δ_g and Δ_h are step sizes in the g and h directions respectively. For numerical stability, we work with $\log(p(\mathbf{M}))$, so that we only require $\log(Z(g, h))$ rather than $Z(g, h)$. Note that although the evaluation of $Z(g, h)$ is relatively costly, it only need be evaluated once for each grid point. Also note that the integral need only be evaluated once for any particular pair $(S(\mathbf{M}), L(\mathbf{M}))$. Hence, the overall computational cost of this approach is not too great. In contrast, when including g and h in the MCMC scheme, $Z(g, h)$ needs to be recalculated every time a new candidate pair (g', h') is proposed, and hence there is also a computational advantage to integrating out these parameters.

4 Examples

We now illustrate the new methodology with two examples, also analyzed by Rodriguez and Schmidler (2010), which have been studied previously in the literature. These are challenging examples, where in each case the proteins have a low sequence identity (percentage of aligned pairs which are the same amino acid residue type), but are structural homologues. In our first example, we analyse one of these pairs, with PDB identification codes 1ACX and 1COB. In our second example, we analyse the pair of proteins with PDB identification codes 1GKY and 2AK3.

It is necessary to specify a value of the volume parameter v . We find that v can have a marked effect on the posterior number of matches, with higher values of v favouring more matches. From experience, we find that a value of $v = 5000$ provides a good starting point in practice, generally giving a reasonable number of matches and sensible solutions (as evidenced, for example, by comparing the resulting alignments with those from the LGA program (Zemla, 2003)). The value of v can then be adjusted to promote more or less matches as desired; in Section 4.3, we further investigate the effect of changing v for both the examples in this section. However, it should be remembered that v is not merely an abstract parameter, but is a parameter in the model which represents the volume of the space in which the (hidden) points lie. Therefore, if real information is known about this, then it could be used to obtain a representative value of v — in our context, this could be the volume of the space which contains the larger of the two observed configurations. In practice, however, it could also be used as a tuning parameter by the user to experiment with a greater/lesser prior propensity for matching.

We initially consider the gap penalty parameters to be fixed, and we use the values $g = 4$ and $h = 0.1$. These are equal to the expected values of g and h from the prior distributions used by Rodriguez and Schmidler (2010), who suggest that a gap opening penalty of the order of 40 times as large as the gap extension penalty is reasonable, following Gerstein and Levitt (1998). In general, it is desirable to penalise the opening of gaps much more than the extension of gaps. For both our examples, we also consider the case where g and h are treated as additional unknown parameters in our model, and apply the methods of Section 3.4.

For the remaining parameters, we use the following settings, unless otherwise stated. We run the MCMC sampler for 4800000 sweeps, with the first 800000 values discarded as burn-in. Inference is made using a thinned sample of 2000 values from the 4000000 post burn-in sample. The prior mean for the translation, $\boldsymbol{\mu}_\tau$, is taken to be the difference between the centroids of the two configurations. Prior information on $\boldsymbol{\tau}$ is weak, so we set $\sigma_\tau = 500$ to give a diffuse prior to reflect this. The prior for the rotation matrix \mathbf{A} is uniform. We set $\alpha = 1$, giving an exponential prior for σ^{-2} with mean $\frac{1}{\beta}$. We keep $\beta = 8$ fixed throughout — posterior inferences are robust to moderate changes of this value. As a starting value for the matching matrix \mathbf{M} , we take the solution of matches given by LGA (Zemla, 2003).

4.1 Example 1

We now give the results from a typical run for our first example, 1ACX (chain A, 108 points) and 1COB (chain A, 151 points). The posterior expectation for the number of matches is 72.04, and the first duplicated match is the 71st most probable, between \mathbf{x}_{27} and \mathbf{y}_{38} with probability 0.42 (the 70th most probable match is between \mathbf{x}_{27} and \mathbf{y}_{39} with probability 0.58). In general, a duplicated match, between \mathbf{x}_j and \mathbf{y}_k say, means that at least one of the indices j or k has already appeared in a match with a higher marginal posterior probability. The marginal posterior probabilities for the matches between individual residues are shown in Figure 2, which clearly highlights regions which are well aligned and regions where there is more uncertainty in the alignment. Also evident are the locations where gaps have been created to enable a better alignment. A diagnostic commonly used in Bioinformatics to measure the quality of an alignment, given \mathbf{A} , $\boldsymbol{\tau}$ and \mathbf{M} , is the root mean squared deviation (RMSD), which is defined as

$$\sqrt{\frac{1}{L} \sum_{j,k:M_{jk}=1} \|\mathbf{x}_j - \mathbf{A}\mathbf{y}_k - \boldsymbol{\tau}\|^2}.$$

For this example, the median RMSD value is 2.06 (corresponding to a solution with 72 matches), with 95% posterior interval (1.89,2.31). Rodriguez and Schmidler (2010) report three solutions, corresponding to a modal value of \mathbf{M} obtained using three different values of a user-specified parameter λ . The three solutions give RMSD values of 2.1 (66 matches), 3.8 (86 matches) and 4.1 (93 matches). Our results appear to be at least comparable to these.

As described in Green and Mardia (2006), the principles of Bayesian decision theory can be used to obtain a posterior point estimate for \mathbf{M} if one is required, by defining a loss function which incorporates costs for falsely declaring matches and missing true matches. It is necessary only to specify a value for a parameter K , where $K = \frac{l_{01}}{l_{01}+l_{10}}$; the term l_{01} denotes the cost incurred for falsely declaring a match, and l_{10} denotes the cost of falsely missing a true match. The point estimate is obtained by minimising the expected loss with respect to the marginal posterior matching probabilities, which can be regarded as a linear assignment problem. Note that larger values of K give fewer matches, since falsely declaring a match incurs a relatively higher cost than missing a true match.

To solve the assignment problem, we use the method of Jonker and Volgenant (1987), and will refer to a particular solution obtained from this algorithm as a linear assignment. A linear assignment with $K = 0.1$ gives 78 matches with RMSD 2.28 and a linear assignment with $K = 0.9$ gives 70 matches with RMSD 1.99 (corresponding to the top 70 matches before the first duplication as described above). The LGA program, with a distance cutoff of 5.0Å, gives 79 matches with RMSD 2.48, which can be used as a guide to suggest that a good solution has been obtained. This appears to be the case, and shows the ability of our method to find a plausible solution, and to explore uncertainty

in the corresponding alignment.

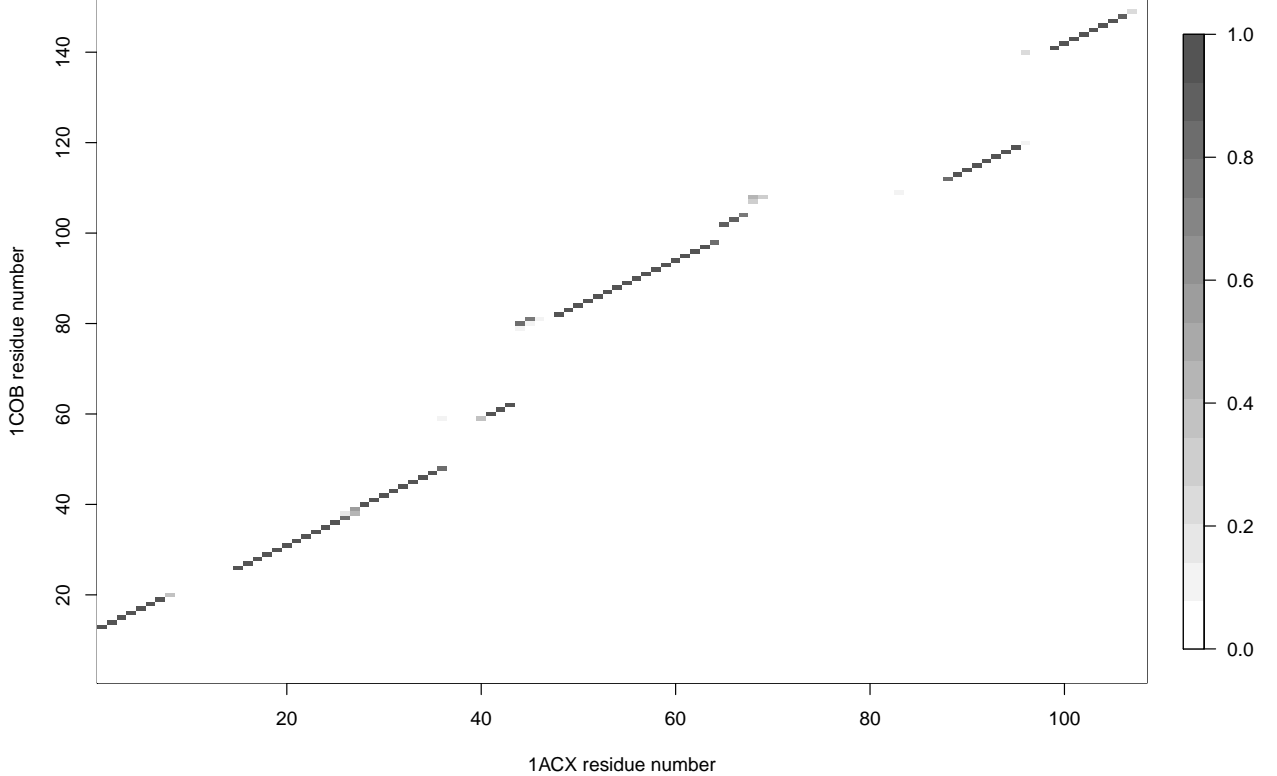


Figure 2: Marginal posterior probabilities for matches between individual pairs of residues for the pair 1ACX-1COB.

4.1.1 Assessing convergence

To ensure that the the method has converged to the main mode, multiple chains were run from different starting values for the parameters \mathbf{A} , $\boldsymbol{\tau}$ and σ , keeping the starting value for the matching matrix \mathbf{M} fixed at the same value. Inspection of the posterior traces of the sampled parameter values suggest good mixing of the sampler, and the RMSD value suggests that a good solution has been obtained. The median value of the log-posterior is 377.9, and the minimum is 338.4. We suppose that the log-posterior monitored in this manner can be used as a good indicator that a suitable solution has been found, and use these values as a benchmark for determining convergence in subsequent runs.

To assess convergence, 10 runs were performed, each using the same settings as previously and with random starting values chosen for the parameters \mathbf{A} , $\boldsymbol{\tau}$ and σ . The

starting values for $\boldsymbol{\tau}$ and σ^{-2} were sampled from their prior distributions, and starting values for the Eulerian angles parameterising \mathbf{A} were drawn uniformly over the range of their possible values.

Nine of the 10 runs had median log-posterior values in the range 377.6 – 378.4 and were considered to have converged to good solutions. The other run had a median log-posterior value of 100.7, which clearly represents convergence to a subsidiary mode and it is therefore discarded. Of the remaining runs, the top 70 most probable matches were the same for each, and the first duplicated match was the 71st most probable in each case. Additionally, a linear assignment with $K = 0.1$ gave the same 78 matches for each of these runs. These results provide strong evidence that the method has converged to a plausible solution, and that we can be confident in the conclusions drawn from the results.

We also implemented a parallel tempering algorithm, designed to alleviate the problem of getting stuck in subsidiary modes. Using this scheme, and starting from arbitrary \mathbf{M} (including the case where there are no initial matches given), the sampler consistently converged to the same mode, corresponding to the solution described above. Hence, we are confident that the results reported correspond to a global dominant mode.

4.1.2 Inference for the gap penalty parameters

We now run the same example, this time allowing updates for g and h , using the hyperparameter values $a_g = 2, b_g = 0.5, a_h = 2, b_h = 20$ as in Rodriguez and Schmidler (2010). These give prior expected values equal to the fixed values for g and h used previously, which were 4 and 0.1 respectively.

The posterior median of g is 3.84, with 95% interval (3.16, 4.68), and the posterior median of h is 0.05, with 95% interval (0.01, 0.14). The prior densities and posterior density estimates are shown in Figure 3. Recall that in the previous analysis, the parameters were fixed as $g = 4, h = 0.1$. These values appear to be compatible with the posterior distributions obtained here, and suggest that these were sensible values to use. In particular, the fixed values are contained within the posterior credible intervals obtained here.

The results obtained for the matches are very similar to those found previously. In particular, the top 70 matches are identical to those found in the previous analysis. The median RMSD was 2.04 (corresponding to a solution with 71 matches), with 95% posterior interval (1.87, 2.28), which again agrees with the previous results.

4.1.3 Integrating out the penalty parameters

We may also consider integrating out g and h , as described in Section 3.4.2. For the integration, we used the ranges $0 \leq g \leq 20$ and $0 \leq h \leq 2$, with evaluation points evenly spaced on a 100×100 grid. These ranges were chosen after careful inspection of the integrand and investigating the performance of the method for a wide range of possible

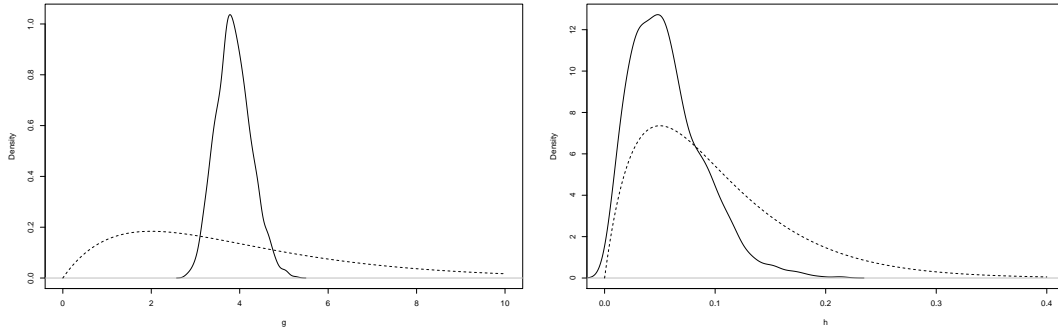


Figure 3: Prior densities and marginal posterior density estimates of g (left) and h (right) for the pair 1ACX-1COB. The dashed lines are the prior densities, and the solid lines the marginal posterior density estimates.

matrices \mathbf{M} . The traces of the posterior samples of $S(\mathbf{M})$ and $L(\mathbf{M})$ are shown in Figure 4. The posterior means of $S(\mathbf{M})$ and $L(\mathbf{M})$ were 16.3 and 100.0 respectively.

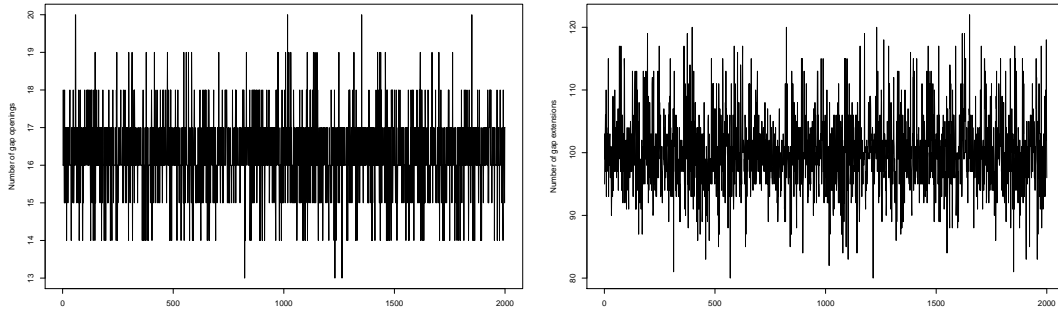


Figure 4: Posterior traces of the number of gap openings (left) and the total number of gap extensions (right).

4.2 Example 2

We now discuss alignment of the pair 1GKY (chain A, 186 points) and 2AK3 (chain A, 226 points). Again, we take the starting alignment of \mathbf{M} as the solution given by LGA, and use the same parameter settings as previously. The marginal probabilities for matches between pairs of points for a typical run are shown in Figure 5. The first duplicated match was the 127th most probable.

A linear assignment on the matches obtained from this run, with $K = 0.2$, gave 131 matches with RMSD 2.25. The median RMSD was 2.20 (corresponding to an alignment with 127 matches), with 95% posterior interval (2.09,2.36). LGA was used to obtain a reference RMSD value to indicate whether this corresponds to a good solution; LGA,

with a distance threshold of 4.0, gave an RMSD of 2.21 with 129 matches, so on this occasion the two methods appear to give similar solutions. Rodriguez and Schmidler (2010) report two RMSD values, corresponding to solutions obtained in two situations (namely, with and without the use of amino acid type information — we discuss using amino acid type information in our model in Section 5). The two RMSD values quoted are 3.5 (without amino acid information) and 1.95 (with amino acid information), but the number of matched points are not given, so it is difficult to interpret these in the context of our results.

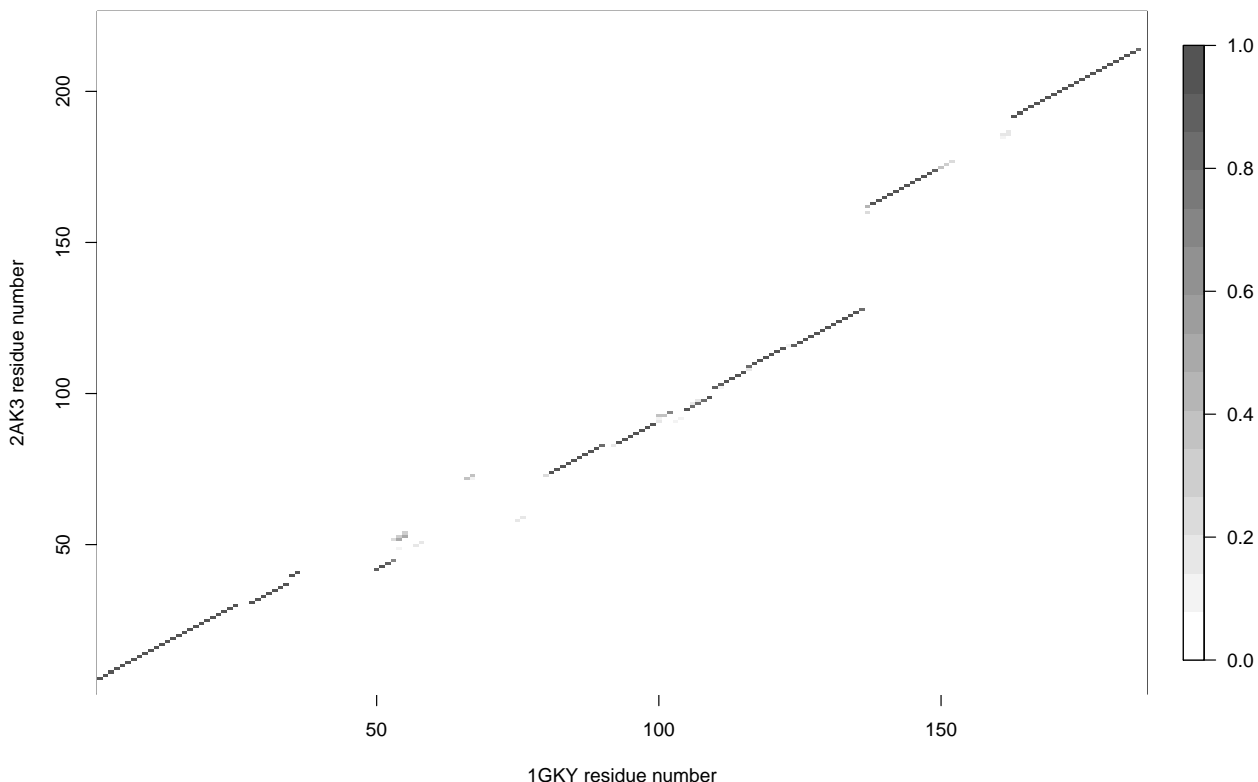


Figure 5: Marginal posterior probabilities for individual pairs of residues for the pair 1GKY-2AK3.

4.2.1 Assessing convergence

To assess convergence, the chains were run 10 times from different starting values of the parameters, apart from \mathbf{M} which was fixed at the same value as above. The median log-posterior value for the initial run described above was 696.9, and as in the previous example, we use this as a diagnostic to suppose that convergence has been reached. Posterior traces, together with the RMSD value, suggest that convergence has been

reached and that the corresponding solution is a good one. For the additional 10 runs, the median log-posterior values were all very similar, ranging from, 696.3 to 697.1, so all were considered to have converged to a good solution. All these runs gave the same top 126 most probable matches, identical to those found in the initial run. This provides strong evidence that convergence has been reached, and that the resulting solution is a plausible one.

4.2.2 Inference for the gap penalty parameters

Again, we can consider the parameters g and h to be quantities about which to draw inference, and sample from their posterior distributions. We use the same prior settings for g and h as in our first example, and the same settings as above for the other parameters. The posterior median for g is 4.33, with 95% posterior interval (3.68,5.05), and the posterior median for h is 0.06, with 95% posterior interval (0.01,0.13). Prior densities and posterior density estimates are shown in Figure 6. The posterior median RMSD is 2.21 (corresponding to a solution with 127 matched points), with 95% posterior interval (2.09,2.36), so for this example the results obtained are virtually identical to the case when g and h were fixed.

Regarding g and h , the results for h are approximately the same as those in the previous example, whereas a higher value of g is suggested here than was the case previously. Again, both posterior intervals contain the fixed values that were used in the previous analysis. The results from both examples suggest that the fixed values of $g = 4$ and $h = 0.1$ are sensible, but that including them as unknowns in the model allows inference about them to be made, enabling subtle changes in their values between different examples to be detected (particularly for the gap opening penalty g).

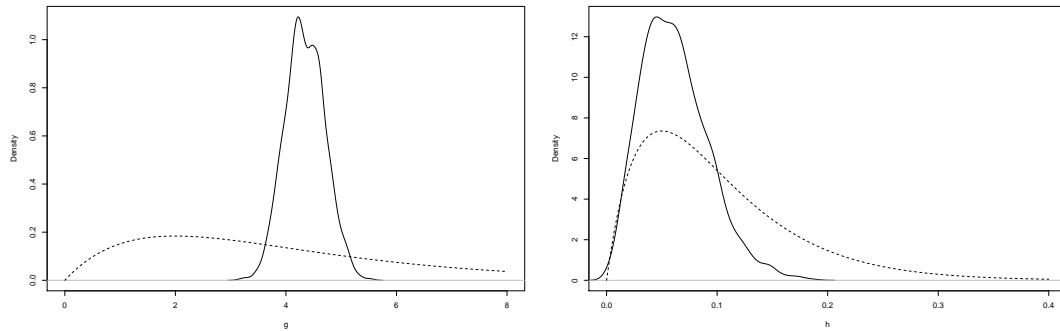


Figure 6: Prior densities and marginal posterior density estimates of g (left) and h (right) for the pair 1GKY-2AK3. The dashed lines are the prior densities, and the solid lines the marginal posterior density estimates.

4.2.3 Integrating out the penalty parameters

Again, we can also consider integrating out g and h . The traces of the posterior samples of $S(\mathbf{M})$ and $L(\mathbf{M})$ are shown in Figure 7. The posterior means of $S(\mathbf{M})$ and $L(\mathbf{M})$ were 20.3 and 136.0 respectively.

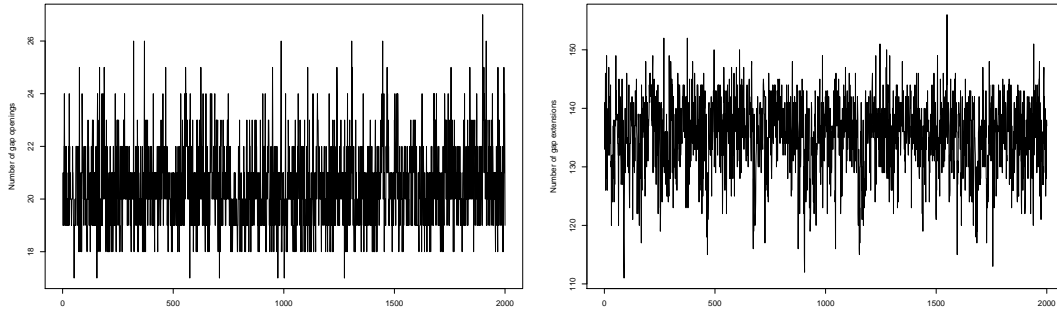


Figure 7: Posterior traces, of the number of gap openings (left) and the total number of gap extensions (right).

4.3 Sensitivity to v

Increasing the value of v gives a greater number of posterior matches for both the examples in this section, as can be seen from the empirical results shown in Tables 1 and 2. As more matches are found, the RMSD also increases accordingly, since the extra matches being added will generally be those which are further apart after transformation, and hence are less likely to be matched; the empirical evidence suggests that a higher value of v is in some sense increasing the matching propensity between points, thereby overcoming the probability barrier associated with large inter-point distances.

A possible explanation is as follows. As v increases, there is an increase in “empty space” which needs to be accounted for in the model. Also, under the Poisson process model for the hidden point locations $\{\mu\}$, an increase in v implies a higher expected number of realisations of points on $\{\mu\}$. Since the number of observed points is fixed and equal to $m + n$, the model can account for this in two ways: either there is a large number of hidden points that do not appear in either of the observed configurations \mathbf{X} or \mathbf{Y} , or more matches can be added between points which have relatively large inter-point distances that are not matched for small v . Both these options are unfavourable under the model, but the empirical evidence from these examples suggests that adding more matches is how the model behaves in practice.

v	Median RMSD (95% posterior interval)	Median L
2000	1.81 (1.65,2.02)	64
5000	2.06 (1.89,2.31)	72
20000	2.55 (2.33,2.78)	82
50000	2.67 (2.48,2.96)	84

Table 1: Posterior RMSD values and number of matches L for different values of the volume v for the pair 1ACX-1COB.

v	Median RMSD (95% posterior interval)	Median L
2000	2.00 (1.82,2.19)	117
5000	2.21 (2.09,2.36)	127
20000	2.47 (2.35,2.78)	141
50000	2.97 (2.68,3.15)	152

Table 2: Posterior RMSD values and number of matches L for different values of the volume v for the pair 1GKY-2AK3.

5 Amino acid information

In addition to using only structural information in the alignment, we can incorporate the amino acid information. At the primary level, a protein is a sequence of letters from an alphabet of 20 letters, with each letter representing one of the 20 amino acids. The amino acid sequences can provide extra information, since some amino acid pairs are more likely to be aligned than others, and measures have been developed to quantify this. The information is often presented in the form of scoring matrices (such as the PAM matrices discussed below), which assign a score to each of the 20×20 pairs of amino acid types which provides a measure of how likely an alignment between any pair of amino acid types is. The scores also take into account the evolutionary distance between two proteins; over longer evolutionary distance, the scoring is more tolerant of alignments between amino acid types that would otherwise be considered unfavourable over a shorter evolutionary distance.

5.1 PAM matrices

The PAM matrices (Dayhoff et al., 1978) are a series of matrices derived to provide a score between any pair of amino acid types, a and b say. The scores are based on the likelihood of a substitution from a to b at any give position on the protein chain over a period of evolution. A substitution is taken to mean a mutation which has spread to the entire protein family. Over long periods of time, multiple mutations may have taken place, so that a substitution from a to b may have taken place via one or more intermediary mutations. The derivation of the PAM matrices uses Markov chain theory, by first estimating the probability of mutations between the different amino acid types

over a short time period (deemed to be one unit of evolutionary time) to give a one-step Markov model. The resulting probability of a substitution from a to b in this period of time is denoted $p_{a,b}$. This model can then be used to estimate substitution probabilities over longer evolutionary timescales.

In general, a PAM- l matrix denotes an evolutionary distance of l units, where l represents the expected percentage of mutations implied by the underlying Markov model over this period. The larger the value of l , the higher the number of expected mutations, implying a longer evolutionary distance. The PAM- l matrices are derived from the PAM-1 transition matrix as follows. From standard Markov chain theory, the probability of observing a mutation from a to b over l units of evolutionary time is $p_{a,b}^{(l)}$. Therefore, the probability of observing an alignment between amino acid types a and b in two sequences which are l evolutionary time units apart is $q_a p_{a,b}^{(l)}$, where q_a is the marginal probability of amino acid type a , which represents the relative abundance of this amino acid in all proteins. The resulting probabilities are then usually rescaled and expressed as log-odds scores for each amino acid pair over an evolutionary distance l . Specifically, the entries in a PAM- l matrix are expressed in the form $C \log \psi_{ab}^l$, where $\psi_{ab}^l = \frac{p_{ab}^{(l)}}{q_a q_b}$ and C is a constant, the choice of which is arbitrary and is not too important — different choices are found in the literature. Therefore, the entries of a PAM matrix give a log-odds score for any pair of amino acid types a and b . The total score of any alignment is then the sum of the individual scores for each aligned pair.

5.2 Sequence likelihood

We now describe the likelihood of the amino acid sequences given an evolutionary distance l . We denote the sequences of the \mathbf{X} and \mathbf{Y} configurations by S^x and S^y respectively. The j th element of S^x is then $s_j^x \in \mathcal{S}$, $j = 1, \dots, m$, where \mathcal{S} is the set of integers 1 – 20, representing each of the 20 amino acid types. A similar definition follows for the n elements of S^y . We then assume the following form for the likelihood of the observed sequences:

$$p(S^x, S^y | \mathbf{M}, \Psi^l) = \prod_{j,k: M_{jk}=1} \psi_{s_j^x s_k^y}^l \prod_{j=1}^m q_{s_j^x} \prod_{k=1}^n q_{s_k^y}, \quad (5)$$

where Ψ^l is a PAM- l matrix as described above.

Assuming that the amino acid sequences are independent of structure, the new joint posterior is obtained by multiplication of equations (3) and (5), giving

$$\begin{aligned} p(\mathbf{M}, \mathbf{A}, \boldsymbol{\tau}, \sigma, \mathbf{x}, \mathbf{y}, S^x, S^y) &\propto p(\mathbf{A})p(\boldsymbol{\tau})p(\sigma)|\mathbf{A}|^n v^L \exp\{-U(\mathbf{M}; g, h)\} \\ &\times \prod_{j=1}^m q_{s_j^x} \prod_{k=1}^n q_{s_k^y} \prod_{j,k: M_{jk}=1} \frac{\psi_{s_j^x s_k^y}^l \phi\{(\mathbf{x}_j - \mathbf{A}\mathbf{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}. \end{aligned} \quad (6)$$

Note that in this expression, we have reverted to the case where the gap penalty parameters g and h are assumed constant. As before, g and h could be considered as unknowns; in this case, the joint posterior would follow in the same way, as the product of equations (4) and (5).

Also note that the term $\prod_{j=1}^m q_{s_j^x} \prod_{k=1}^n q_{s_k^y}$ is a constant with respect to \mathbf{M} , so all relevant amino acid type information about \mathbf{M} is contained in the matched points, through the term $\prod_{j,k:M_{jk}=1} \psi_{s_j^x s_k^y}^l$.

If the value of l is considered as known and kept fixed, then the same PAM matrix Ψ^l is used throughout. In Section 5.3 below, we describe a method for estimating l , by including it as an extra unknown parameter in our model.

5.3 Inference for evolutionary distance

As described previously, a PAM- l matrix corresponds to l expected mutations, and larger values of l imply a longer evolutionary distance. Treating the value of l as fixed implies knowledge of the evolutionary distance, and a PAM-250 matrix is commonly used when aligning sequences thought to be distantly related. In principle, any PAM matrix may be used if there is prior knowledge of the evolutionary distance. However, the Bayesian approach naturally allows us to incorporate uncertainty as to the evolutionary distance between two proteins, by including l as an unknown parameter in the model. Therefore, as a by-product of the alignment we can also obtain a natural measure of the evolutionary distance. The adjustment to the model to incorporate l , and the method for sampling l from the full posterior distribution, is now described. The method is similar to that of Rodriguez and Schmidler (2010), although we use a non-uniform prior distribution for l . Challis and Schmidler (2012) have modelled the evolutionary process directly, using diffusions to model the evolution of structures, as their primary interest was inference for the evolutionary process.

We consider 37 possible values of the PAM distance l in the set $\mathcal{L} = \{40, 50, 60, \dots, 380, 390, 400\}$. A-priori, we assume that l has a normal distribution with mean μ_l and variance σ_l^2 ; we then use a discretized version of this distribution to obtain a prior probability mass function over the set of PAM distances considered. Specifically, for all $l \in \mathcal{L}$, we assign a prior weight p_l , with

$$p_l = \Phi\left(\frac{l + 5 - \mu_l}{\sigma_l}\right) - \Phi\left(\frac{l - 5 - \mu_l}{\sigma_l}\right),$$

where $\Phi(\cdot)$ is the distribution function of the standard normal distribution. These weights could of course be normalized to give a true probability mass function, but in our implementation we only require the unnormalized weights. As a default, we could set $\mu_l = 250$, as PAM-250 matrices are often used in alignments of protein se-

quences thought to be distantly related. If extra information is available, this can be incorporated by altering μ_l and σ_l accordingly.

To sample from the posterior distribution for l we use Metropolis updates as follows. Suppose the current PAM distance at any point in the MCMC run is l ; we can then propose a move to a new PAM distance, l' say, and accept this with probability

$$\alpha_l = \min \left[1, \frac{p(S^x, S^y | \mathbf{M}, \Psi^{l'}) p_{l'}}{p(S^x, S^y | \mathbf{M}, \Psi^l) p_l} \right] = \min \left[1, \frac{p_{l'}}{p_l} \prod_{j,k: M_{jk}=1} \frac{\psi_{s_j^x s_k^y}^{l'}}{\psi_{s_j^x s_k^y}^l} \right],$$

assuming the proposal distribution is symmetric. We draw the proposal value l' from a uniform distribution over the set of possible PAM distances, so the ratio of proposal densities is 1.

5.4 Examples

We now illustrate the method for modelling evolutionary distance with some examples. We keep the gap penalty parameters g and h fixed, with $g = 4$ and $h = 0.1$, and the other parameter settings are as previously. We fix $\mu_l = 250$ as a prior mean for the PAM distance and consider various values of σ_l .

The first example is the yeast guanylate kinase (1GKY) analyzed previously, with a mouse guanylate kinase (1LVG). A BLAST (Altschul et al., 1990) search returns a highly significant sequence alignment, with 51.9% sequence identity. (BLAST is a sequence alignment method, and sequence identity is the percentage of aligned residues of the same amino acid type.) This is a high value for sequence identity, suggesting that the two sequences are closely related. We first consider the case with $\sigma_l = 100$, so that approximately 90% of the prior mass for the PAM distance is contained in the interval [120,380]. The posterior distribution of the PAM distance is shown in the top-left panel of Figure 8, which indicates a short evolutionary distance as would be expected (modal value PAM-80). Acceptance rates for the PAM matrix proposals were around 5%. The posterior modal PAM distance is very different to the mean of the prior, suggesting that the data are very informative. Only when we start to reduce σ_l does the posterior mode start to shift towards the prior mean (Figure 8 top-right and bottom), but even when $\sigma_l = 50$, so that approximately 90% of the prior mass for the PAM distance is contained in the interval [170,330], the data still dominate and the modal posterior PAM distance is still 80.

For our second example, we consider the pair 1GKY-2AK3 analyzed previously. These proteins show little sequence identity, but are known to possess structural similarities. Again we use $\mu_l = 250$, and begin by using $\sigma_l = 100$. The acceptance rate for proposed updates for l in this example were around 30%. The top-left panel of Figure 9 shows the posterior distribution of the PAM distance. The marginal posterior distribution of l is unimodal, with a modal value of 260, suggesting a longer evolutionary distance between the two proteins than in the previous example. The posterior mean number of

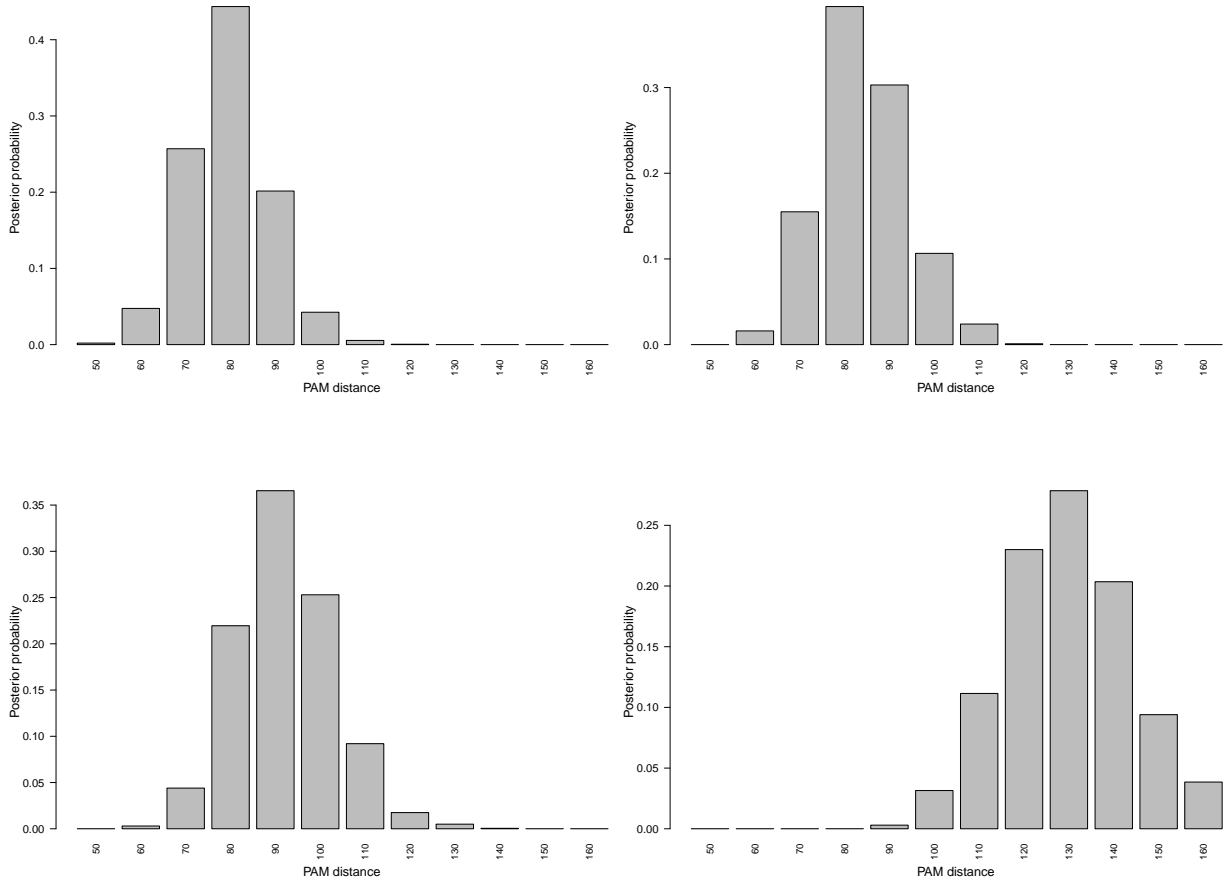


Figure 8: Posterior distribution of PAM distance for 1GKY-1LVG. Top: $\sigma_l = 100$ (left), $\sigma_l = 50$ (right). Bottom: $\sigma_l = 35$ (left), $\sigma_l = 20$ (right).

matches is 127.7. In this example, the posterior mode is close to the prior mean even for a large value of σ_l , but as σ_l decreases the posterior distribution again moves towards, and is more concentrated around, the prior mean of 250 as we would expect (Figure 9 top-right and bottom).

Zhu et al. (1998) developed a method for modelling evolutionary distance, based on alignment of sequences alone, and they too considered the pair 1GKY-2AK3. These authors report a multimodal posterior distribution for the PAM distance, with modes at 110, 140 and 200. The same example has also been considered by Rodriguez and Schmidler (2010), who obtain a unimodal posterior distribution much like ours, but with the mode at PAM-210. The differences could be down to parameter settings; for instance, we find the posterior distribution for the PAM distance is a little sensitive to the volume parameter v , which, as discussed previously, strongly influences the posterior number of matches — here we have kept the value of v fixed at 5000. The authors do not give details of the number of matched points in the alignment giving rise to their given estimate of PAM distance; the two methods may well give similar results for a

comparable number of matches, or there may be true differences between the two.

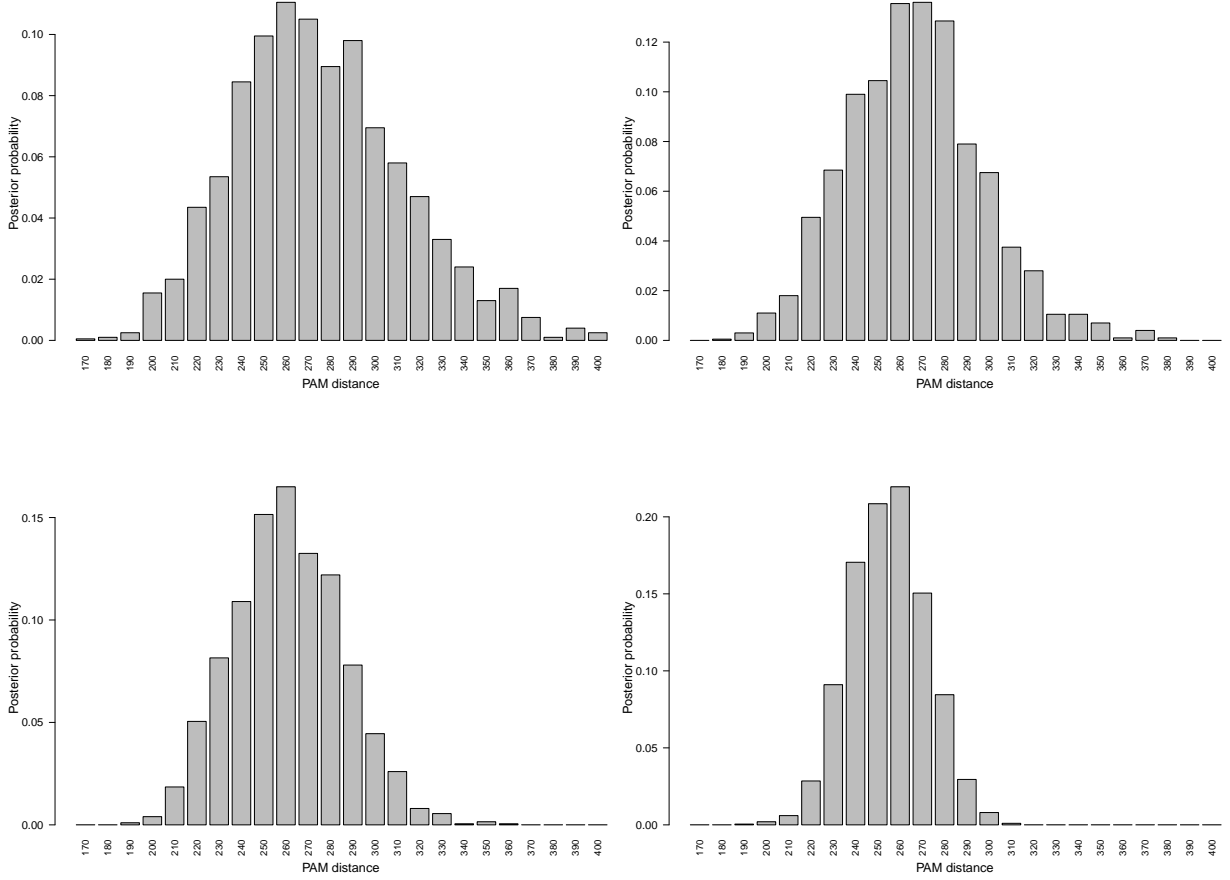


Figure 9: Posterior distribution of PAM distance for 1GKY-2AK3. Top: $\sigma_l = 100$ (left), $\sigma_l = 50$ (right). Bottom: $\sigma_l = 35$ (left), $\sigma_l = 20$ (right).

As mentioned above, the posterior distribution of the PAM distance is influenced by the volume parameter v . In particular, increasing v promotes longer evolutionary distances, as can be seen in Figure 10 below. This is because as v increases, more matches are being accepted, including more of those which would previously have had a low probability of matching due to incompatibility in their amino acid types. This suggests longer evolutionary distances in order to account for matches between incompatible amino acid types, since the PAM model becomes more tolerant to such matches as the PAM number increases. Figure 10 shows the posterior distribution of PAM matrices for $v = 20000$ and $v = 50000$. As v increases, the number of matches increases (a posterior mean number of matches of 139.68 and 150.13 for $v = 20000$ and $v = 50000$ respectively). For the case $v = 50000$, the posterior mode has increased to PAM-300, and the higher PAM values generally have greater posterior probability. This suggests that more matches between residues with incompatible amino acid types have been added, which were previously very unlikely for lower values of v , thus forcing higher values of evolutionary distance to account for them. In contrast, when $v = 1000$ the

posterior mode of PAM distance is 240, with mean number of matches 100.4.

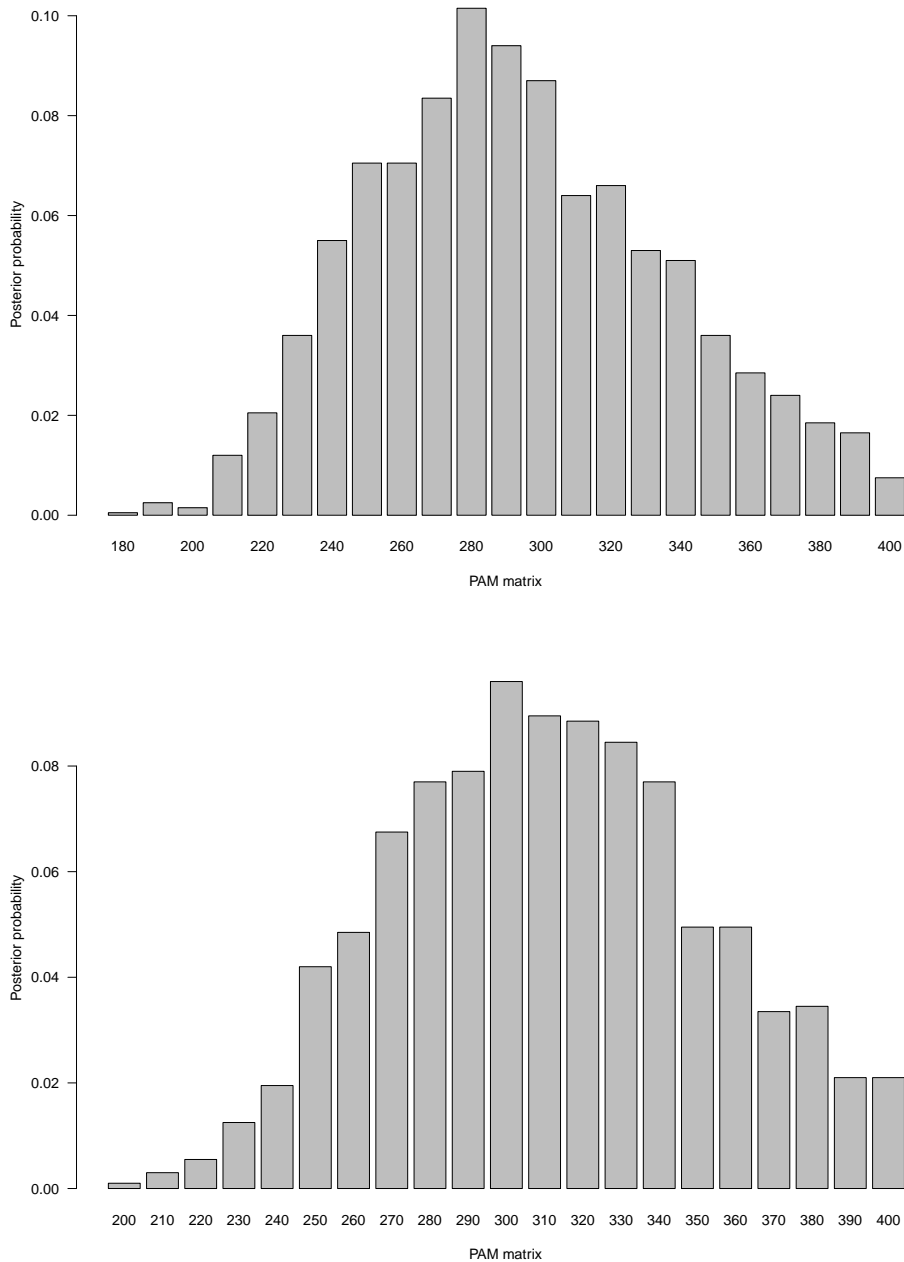


Figure 10: Posterior distribution of PAM matrices for the pair 1GKY-2AK3, with $v = 20000$ (top) and $v = 50000$ (bottom).

6 Discussion

In this paper we have presented a fully Bayesian model for the alignment of protein structures. The model is based on the model of Green and Mardia (2006), but accounts for the constraint that the sequence ordering of the points in each configuration is meaningful and must be preserved when matching pairs of points. This required a new prior model for the matching matrix \mathbf{M} . We have used a particular gap penalty function widely used when performing sequence alignment, but other penalty functions could be used which satisfy our formulation of the total penalty as a sum over individual contributions from each pair of matched indices. Rodriguez and Schmidler (2010) have also developed a Bayesian model for protein structure alignment; we have used the same prior model for \mathbf{M} , but their method of sampling alignments from the posterior distribution is quite different to ours, in that an entire new alignment is sampled at each iteration as opposed to the small perturbations of our proposals. Additionally, the authors optimise over the registration parameters, which can be viewed as using a Laplace approximation to the marginal posterior distribution. We have treated the registration parameters as additional unknown parameters about which to draw inference, and sampling them from the posterior allows us to account for the extra uncertainty in the alignment as a result of the uncertainty in these parameters. We note that Kenobi and Dryden (2012) have begun numerical comparisons between the two approaches in a particular situation, namely where rigid-body transformations are used and no sequence order constraint is imposed. The flexibility of the fully Bayesian method to handle different transformations and constraints has been further illustrated in this paper and the recent papers by Mardia et al. (2013) and Forbes and Lauritzen (2013).

We have illustrated our method on challenging examples seen previously in the literature, and have shown that our method finds sensible solutions with low RMSD and a high number of matches relative to other methods. Our method also allows amino acid information to be easily incorporated, and we have shown that this leads to a natural method for estimating the evolutionary distance between two proteins, by allowing inference for an additional parameter representing evolutionary distance.

References

- Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410.
- Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, N. E. (2000). The protein data bank. *Nucleic Acids Research*, 28:235–242.
- Challis, C. J. and Schmidler, S. C. (2012). A stochastic evolutionary model for protein structure alignment and phylogeny. *Molecular Biology and Evolution*, 29:3575–3587.

- Dayhoff, M. O., Schwartz, R., and Orcutt, B. C. (1978). A model of evolutionary change in proteins. *Atlas of protein sequence and structure*, 5:345–358.
- Dryden, I. L., Hirst, J. D., and Melville, J. L. (2007). Statistical analysis of unlabeled point sets: comparing molecules in chemoinformatics. *Biometrics*, 63:237–251.
- Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge.
- Forbes, P. G. M. and Lauritzen, S. (2013). Fingerprint analysis using Bayesian alignment. In Mardia, K. V., Gusnato, A., Riley, A. D., and Voss, J., editors, *LASR 2013 — Statistical Models and Methods for non-Euclidean Data with Current Scientific Applications*, pages 81–84. Leeds University Press, Leeds.
- Gerstein, M. and Levitt, M. (1998). Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins. *Protein Science*, 7:445–456.
- Green, P. J. and Mardia, K. V. (2006). Bayesian alignment using hierarchical models, with applications in protein bioinformatics. *Biometrika*, 93(2):235–254.
- Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89:10915–10919.
- Holm, L. and Sander, C. (1993). Protein structure comparison by alignment of distance matrices. *Journal of Molecular Biology*, 233:123–138.
- Jonker, R. and Volgenant, A. (1987). A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38:325–340.
- Kenobi, K. and Dryden, I. L. (2012). Bayesian matching of unlabeled point sets using procrustes and configuration models. *Bayesian Analysis*, 7:547–566.
- Liu, J. S. and Lawrence, C. E. (1999). Bayesian inference on biopolymer models. *Bioinformatics*, 15:38–52.
- Mardia, K. V., Fallaize, C. J., Barber, S., Jackson, R. M., and Theobald, D. L. (2013). Bayesian alignment of similarity shapes. *The Annals of Applied Statistics*, 7:989–1009.
- Orengo, C. A. and Taylor, W. R. (1996). Ssap: sequential structure alignment program for protein structure comparison. *Methods in Enzymology*, 266:617–635.
- Oritz, A. R., Strauss, C. E. M., and Olmea, O. (2002). Mammoth (matching molecular models obtained from theory): An automated method for model comparison. *Protein Science*, 11:2606–2621.
- Rodriguez, A. and Schmidler, S. (2010). Bayesian protein structural alignment. *online*, <http://www.stat.duke.edu/~scs/Publications.shtml>.

- Schmidler, S. C. (2007). Fast Bayesian shape matching using geometric algorithms. In Bernardo, J. M., Bayarri, J., Berger, J. O., Dawid, A. P., Heckerman, D., Smith, A. F., and West, M., editors, *Bayesian Statistics 8*, pages 471–490. Oxford University Press, Oxford.
- Shindyalov, I. N. and Bourne, P. E. (1998). Protein structure alignment by incremental combinatorial extension (ce) of the optimal path. *Protein Engineering design and selection*, 11:739–747.
- Wilkinson, D. J. (2007). Discussion of “Fast Bayesian shape matching using geometric algorithms”. In Bernardo, J. M., Bayarri, J., Berger, J. O., Dawid, A. P., Heckerman, D., Smith, A. F., and West, M., editors, *Bayesian Statistics 8*, pages 483–487. Oxford University Press, Oxford.
- Zemla, A. (2003). LGA: a method for finding 3d similarities in protein structures. *Nucleic Acids Research*, 31:3370–3374.
- Zhu, J., Liu, J. S., and Lawrence, C. E. (1998). Bayesian adaptive sequence alignment algorithms. *Bioinformatics*, 14:25–39.